UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/006,551	11/30/2001	Christopher D.S. Donham	NVIDP064/P000286	2643

28875 7590 06/26/2008

Zilka-Kotab, PC
P.O. BOX 721120
SAN JOSE, CA 95172-1120

EXAMINER	
AMIN, JWALANT B	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
06/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/006,551
Filing Date: November 30, 2001
Appellant(s): DONHAM ET AL.

Kevin Zilka
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/19/2008 appealing from the Office action mailed 9/20/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The following are the related appeals, interferences, and judicial proceedings known to the examiner which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

Appeal brief filed on 5/4/2004

Appeal brief filed on 1/18/2006

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

US 5,987,567	Rivard	11-1999
US 5,831,640	Wang	11-1998

AAPA "Applicant Admitted Prior Art "

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 1-12, 18-21, 24-28 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rivard (US 5,987,567), and in view of Wang (5,831,640).

2. Regarding claims 1, 21, 24-27 and 30, Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches a method for execution with a system including a tangible computer readable medium, the method for retrieving instructions from video memory (DRAM 655) utilizing a texture module (texture mapping stage 645 and texel cache system 650 combined together corresponds to texture module) in a graphics pipeline (graphics pipeline 640), comprising

sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval (the examiner interprets that memory requests/read requests for all

misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information; the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; here DRAM sends memory data based on the memory requests/read requests from the texture module. Memory requests/read requests act as an instruction to DRAM, which performs a particular function based on the request); and

receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module).

Although Rivard discloses the limitations as stated above, Rivard does not explicitly teach to combine texture mapping stage and texel cache system to form a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module. The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to

send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions). Although Rivard discloses the limitations as stated above, Rivard does not explicitly teach that the memory returns instructions along with data, in response to instruction request from the texture module. However, Wang teaches a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

3. Regarding claim 2, Rivard teaches a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, comprising sending a texture request (texture lookup requests) to memory utilizing the texture module in the graphics pipeline (Fig. 6, col. 6 lines 50-60).

4. Regarding claim 3, Rivard teaches receiving texture information (memory data) from the memory in response to the texture request utilizing the texture module in the graphics pipeline (Fig. 6, col. 6 lines 50-56).

5. Regarding claim 4, Rivard teaches the video memory includes a frame buffer (col. 3 lines 5-10).

6. Regarding claim 5, Rivard teaches the memory includes direct random access memory (DRAM) (col.4 lines 32-35 and lines 45-55).

7. Regarding claim 6, Rivard teaches the instructions are adapted for controlling a texture environment module (pipeline latency elements) coupled to the texture module (Fig.6, Fig.10).

8. Regarding claim 7, Rivard teaches the instructions control the manner in which the texture environment module processes the texture information (col.7 lines 3-7).

9. Regarding claim 8, Rivard teaches receiving initial instructions from a rasterizer module (graphic accelerator having graphic pipeline stage shows that the texture module receives data/instructions from the upper modules of the pipeline) coupled to the texture module (Fig.6).

10. Regarding claim 9, Rivard teaches the initial instructions control at least the sending of the instruction request by the texture module (Fig. 6; graphic accelerator

having graphic pipeline stage shows that the texture module receives data/instructions from the upper modules of the pipeline; thus based on these instructions, the texture module sends the instruction request to the DRAM).

11. Regarding claim 10, Rivard teaches temporarily storing the instructions and the texture information in cache (cache data store and memory data resolver) (Fig. 6, Fig. 10).

12. Regarding claim 11, Rivard teaches the cache is resident on the texture module (Fig.6, Fig.10).

13. Regarding claim 12, Rivard teaches that each piece of texture information and each of the instructions are of a similar size in the memory (col. 6 lines 50-67 and col. 7 lines 1-15; texture mapping stage includes the ALU which reformats data including resizing of texel data types for uniformity).

14. Regarding claim 18, although Rivard teaches the above-discussed limitations, Rivard does not explicitly teach a complete instruction set is received in response to the instruction request. However, Wang (col. 5 lines 43-67, col. 6 lines 1-47) teaches a graphics subunit (texture module) of a graphics hardware system executing a series of display instructions (set of instructions) stored in computer memory. The graphics subunit receives display instructions including texture data based on the supplied data and control signals (the single display instruction received by the graphics from the sub-routine process in response to the control signals correspond to a complete set of instruction; this instruction in the sub-routine is executed by the graphics subunit for rendering the concerned graphics primitive, and therefore it is considered as a complete

set of instruction). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return a complete set of instructions as taught by Wang to the texture module of Rivard because this instruction set is needed to render the concerned graphics primitive (col. 5 lines 43-45 and lines 63-67).

15. Regarding claims 19, although Rivard teaches the above-discussed limitations, Rivard does not explicitly teach a partial instruction set is received in response to the instruction request. However, Wang (col. 5 lines 43-67, col. 6 lines 1-47) teaches a graphics subunit (texture module) of a graphics hardware system executing a series of display instructions (set of instructions) stored in computer memory. The graphics subunit receives display instructions including texture data based on the supplied data and control signals (the single display instruction received by the graphics from the sub-routine process in response to the control signals correspond to a partial set of instruction; this instruction in the sub-routine is executed by the graphics subunit for rendering the concerned graphics primitive, but other instructions are needed for rendering other primitives, and therefore overall this single instruction is considered as a partial set of instruction). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

16. Regarding claim 20, the statements presented above, with respect to claims 1 and 19, are incorporated herein.

17. Regarding claim 28, Rivard teaches a texture mapping stage as shown in Fig. 6 receiving data (instructions) from the rasterizer module of the pipeline and then receiving data (instructions) from the DRAM. Thus the instructions received from the DRAM via the texel cache system are considered to be the additional instructions. Please refer to the rejection of claims 1, 2, 3, 8 and 10 for further details regarding the rejection of other limitations.

18. Claims 13-17, 22, 23, 29, are rejected under 35 U. S.C. 103(a) as being unpatentable over Rivard and Wang, and further in view of Applicant Admitted Prior Art (AAPA).

19. Regarding claim 13, although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach controlling the texture module utilizing a shader module coupled thereto. However, AAPA teaches controlling the texture module utilizing a shader module coupled thereto (Fig.3; shader module is also coupled to the rasterizer module). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to incorporate the shading module of AAPA into the texture module of Rivard and Wang because combination of shading module and texture module would enable a shading function to the graphic pipeline.

20. Regarding claim 14, although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach the shader module controls the sending

of instruction request and texture request by the texture module. However, AAPA teaches this limitation (Fig. 3 page 5 lines 24-31).

21. Regarding claim 15, although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach the shader module processes a plurality of pixels with the texture information based on the instructions. However, AAPA teaches this limitation (Fig. 3).

22. Regarding claim 16, although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach the shader module is capable of reusing the texture information in order to request further texture information from the video memory (control the looping of texture process). However, AAPA teaches this limitation (Fig. 3 page 4 lines 24-31).

23. Regarding claim 17, although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach ceasing the processing upon the receipt of terminate instruction (require significant amount of time to push down the pipeline). However, AAPA teaches this limitation (Fig. 3 page 5 lines 7-15).

24. Regarding claims 22, 23, although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach the texture module is adapted for operating in a plurality of different modes. However, AAPA teach texture module is

adapted for operating in a plurality of different modes. See page 3 lines 20-25. It would have been obvious to a person of ordinary skill in the art at the time of the invention to incorporate texture module of Rivard into the texture module of AAPA because a combination of texture module operating in plurality of difference modes and the texture module of Rivard would provide components of the texture module processing the texels in various ways such as an address calculation module allow various dimensionality textures.

25. Regarding claim 29, the statements presented above, with respect to claims 1, 2, 3, 8, 10, 13, 14, 15, 17 and 28 are included herein.

(10) Response to Argument

The examiner will like to point that some of the arguments repeated throughout the appeal brief are similar in scope and therefore, will be addressed together.

- Regarding claims 1-12, 21, 24 and 27, appellant argues that "… there is no suggestion that the modification of the Rivard reference is desired" (see pg. 13 of appeal brief). Appellant further argues "… modifying the Rivard reference, would change the principle of operation of the Rivard system, since it would obviate the need for the purposefully included pipeline latency elements 1025" (see pg. 13-14 of appeal brief).

However, the examiner interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work

together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element;

however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). Although Rivard discloses the limitations as stated above, Rivard does not explicitly teach that the memory returns instructions along with data, in response to instruction request from the texture module. However, Wang teaches a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that there is nothing in the teaching of either Rivard or Wang that teaches against such a modification of Rivard's system). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claims 1-12, 21, 24 and 27, appellant argues that "... Rivard does not recognize one of the various possible problems solved by appellant namely to accommodate the programmability of recent texture and shader modules without being inhibited by the size of associated programs" (see pg. 14).

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., to accommodate the programmability of recent texture and shader modules without being inhibited by the size of associated programs) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Moreover, it should be noted that Rivard does not need to teach any limitations that is not claimed by appellant, and also Rivard does not need to solve similar problems as solved by appellant's claimed invention.

- Regarding claims 1-12, 21, 24 and 27, appellant argues that disclosing that a memory request is generated for all misses, as in Rivard, fails to teach "sending an instruction request to video memory, where a texture module … sends the instruction request to the video memory" (see pg. 17). Appellant further argues that Rivard's disclosure that the memory request generator 120, which is included in the texel cache system generates memory requests for all misses and forwards to DRAM for information retrieval, fails to teach " sending an instruction request to video memory, where a texture module … sends the instruction request to the video memory" (see pg. 18, 20 and 22-23). Appellant further argues "... memory data in Rivard simply relates to

texel data, which fails to suggest sending an instruction request to video memory, where a texture module … sends the instruction request to the video memory" (see pg. 19). Appellant further argues that the texture mode indicating a texture lookup, in addition to the memory request generator performing a memory request, as in Rivard, simply fails to suggest "... sending an instruction request to video memory, where a texture module ... sends the instruction request to the video memory" (see pg. 21). Appellant further argues that transferring graphical information into DRAM and outputting cached texture values, as in Rivard, fails to suggest "sending an instruction request to video memory, where a texture module ... sends the instruction request to the video memory" (see pg. 22).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because

it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

- Regarding claims 1-12, 21, 24 and 27, appellant argues "… the texture mapping stage is separate from the texture cache system, as disclosed in Rivard and thus fails to suggest sending an instruction request to video memory, where a texture module … sends the instruction request to the video memory" (see pg. 17-18). Appellant further argues that "… Rivard does not explicitly teach to combine texture mapping stage and texel cache system to form a texture module" (see pg. 23 and 29-30).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video

memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read

request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

- Regarding claims 1-12, 21, 24 and 27, appellant further argues "… Rivard's memory request to the DRAM fails to suggest an instruction request" (see pg. 18).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is further interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and

can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

- Regarding claims 1-12, 21, 24 and 27, appellant argues that "… coordinating the arrival of memory data where the data is conveniently aligned in the texture map, as in Rivard, fails to teach sending an instruction request" (see pg. 19).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an

instruction to DRAM based on which DRAM performs a particular function. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and

moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions).

- Regarding claims 1-12, 21, 24 and 27, appellant further argues "... generating and forwarding a memory request to DRAM, as in Rivard, fails to suggest sending an instruction request where instructions are received in response to the instruction request" (see pg. 20 and 27). Appellant also argues that returning memory data which is stored at a write address after a cache miss, as in Rivard, fails to meet, "receiving instructions from the video memory in response to the instruction request" (see pg. 24), and that Rivard's memory data fails to teach or suggest "instructions" (see pg. 24). Appellant further argues that forwarding memory requests to DRAM for information retrieval, as in Rivard, fails to teach "receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline" (see pg. 26-28).

Appellant further raises similar argument that a graphics subunit executing instructions stored in computer memory, as in Wang, fails to suggest "receiving instructions from the video memory in response to the instruction request utilizing the texture module" (pg. 28-29).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208

USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In this case, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for

further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of

ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and

data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claims 1-12, 21, 24 and 27, appellant further argues that performing memory requests before instruction information reaches the cache data store, as in Rivard, in addition to the disclosure of executing display instructions that may include texture data, as in Wang, fails to suggest "receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline" (see pg. 26-27). Appellant further argues "… it would not be obvious from the teachings of Rivard and Wang to receive instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline" (see pg. 27).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The

examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools

necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a

constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further

processing, where the display instructions include texture data). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claims 25-26, appellant argues that disclosing that a memory request is generated for all misses, as in Rivard, fails to teach “sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 31). Appellant further argues that Rivard’s disclosure that the memory request generator 120, which is included in the texel cache system generates memory requests for all misses and forwards to DRAM for information retrieval, fails to teach " sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 32, 36, 37). Appellant further argues "... memory data in Rivard simply relates to texel data, which fails to suggest sending an instruction request to video memory, where a texture module … sends the instruction request to the video memory” (see pg. 33). Appellant further argues that the texture mode indicating a texture lookup, in addition to the memory request generator performing a memory request, as in Rivard, simply fails to suggest “... sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 35). Appellant further argues that transferring graphical information into DRAM and outputting cached texture values, as in Rivard,

fails to suggest "sending an instruction request to video memory, where a texture module ... sends the instruction request to the video memory" (see pg. 36).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claims 25-26, appellant argues "… the texture mapping stage is separate from the texture cache system, as disclosed in Rivard and thus fails to suggest sending an instruction request to video memory, where a texture module … sends the instruction request to the video memory" (see pg. 31).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claims 25-26, appellant further argues "… Rivard's memory request to the DRAM fails to suggest an instruction request" (see pg. 32, 37).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claims 25-26, appellant argues that "… coordinating the arrival of memory data where the data is conveniently aligned in the texture map, as in Rivard, fails to teach sending an instruction request" (see pg. 33).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claims 25-26, appellant further argues "... generating and forwarding a memory request to DRAM, as in Rivard, fails to suggest sending an instruction request where instructions are received in response to the instruction request" (see pg. 34, 38). Appellant also argues that returning memory data to a cache data store, as in Rivard, fails to meet, "receiving instructions from the video memory in response to the instruction request" (see pg. 37-38), and that Rivard's memory data fails to teach or suggest "instructions" (see pg. 38). Appellant further argues that forwarding memory requests to DRAM for information retrieval, as in Rivard, fails to teach "receiving instructions from the video memory in response to the instruction request" (see pg. 40).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claims 25-26, appellant further argues that performing memory requests before instruction information reaches the cache data store, as in Rivard, in addition to the disclosure of executing display instructions that may include texture data, as in Wang, fails to suggest "receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline" (see pg. 40). Appellant further argues "… it would not be obvious from the teachings of Rivard

and Wang to receive instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline" (see pg. 41).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claim 28, appellant argues that disclosing that a memory request is generated for all misses, as in Rivard, fails to teach “sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 42). Appellant further argues that performing memory requests before instruction information reaches the cache data store, as in Rivard, fails to teach “sending an instruction request” (see pg. 42). Appellant further argues that Rivard’s disclosure that the memory request generator generating a memory request and forwarding the request to DRAM for information retrieval, fails to teach " sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 43, 45, 46). Appellant further argues that the texture mode indicating a texture lookup, in addition to the memory request generator performing a memory request, as in Rivard, simply fails to suggest “... sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 44). Appellant further argues that transferring graphical information into DRAM and outputting cached texture values, as in Rivard, fails to suggest “sending an instruction request to video memory, where a texture module sends the instruction request to the video memory” (see pg. 45).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claim 28, appellant further argues "… Rivard's memory request to the DRAM fails to suggest an instruction request" (see pg. 46).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claim 28, appellant argues that merely storing memory data at the write address, as in Rivard, fails to suggest "receiving additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 46). Appellant further argues "… the texture mapping stage is separate from the texture cache system, as disclosed in Rivard and thus fails to suggest receiving additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 47). Appellant also argues that "… coordinating the arrival of memory data where the data is conveniently aligned in the texture map, as in Rivard, fails to teach receiving additional instructions" (see pg. 47). Appellant further argues that forwarding memory requests to DRAM for information retrieval, as in Rivard, fails to teach "receiving additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 49).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections

are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In this case, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction

requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of

ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module). Rivard also teaches a texture mapping stage as shown in Fig. 6 receiving data (instructions) from the rasterizer module of the pipeline and then receiving data (instructions) from the DRAM (it should be noted that the data has it's associated instructions; texture mapping stage in the graphics pipeline receives data from the rasterizer module to perform texturing/rendering operation; this data received from the rasterizer module is considered to be primary data). Therefore, the data received from the DRAM via the texel cache system is considered to be the additional data (Wang reference as shown below teaches that the instructions include data).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated

instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return

display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that the instructions and data returned by the display list to the graphics hardware system is considered additional instructions or data as the Rivard reference already showed that a rasterizer module passes on preliminary instructions or data to the texture module; it should also be noted that the graphics hardware system includes a 3D graphics subunit that includes a texture engine that is responsible for retrieving the texture map data for the polygon and mapping the texels of the texture data onto the pixels of the polygon; this texture engine of Wang is functional equivalent of the texture module as suggested by Rivard). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 28, appellant further argues that performing memory requests before instruction information reaches the cache data store, as in Rivard, in addition to the disclosure of executing display instructions that may include texture data, as in Wang, fails to suggest “receiving additional instructions from the video memory in response to the instruction request utilizing the texture module” (see pg. 49). Appellant further argues “… it would not be obvious from the teachings of Rivard and Wang to receive additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 50).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given

to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module). Rivard also teaches a texture mapping stage as shown in Fig. 6 receiving data (instructions) from the rasterizer module of the pipeline and then receiving data (instructions) from the DRAM (it should be noted that the data has it's associated instructions; texture mapping stage in the graphics pipeline receives data from the rasterizer module to perform texturing/rendering operation; this data received from the rasterizer module is considered to be primary data). Therefore, the data received from the DRAM via the texel cache system is considered to be the additional data (Wang reference as shown below teaches that the instructions include data).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and

data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that

the instructions and data returned by the display list to the graphics hardware system is considered additional instructions or data as the Rivard reference already showed that a rasterizer module passes on preliminary instructions or data to the texture module; it should also be noted that the graphics hardware system includes a 3D graphics subunit that includes a texture engine that is responsible for retrieving the texture map data for the polygon and mapping the texels of the texture data onto the pixels of the polygon; this texture engine of Wang is functional equivalent of the texture module as suggested by Rivard). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 30, appellant argues that disclosing that a memory request is generated for all misses, as in Rivard, fails to teach “sending an instruction request to video memory, where a cache in the graphics pipeline sends the instruction request to the video memory” (see pg. 52). Appellant further argues that Rivard’s disclosure that the memory request generator generates a memory request for all misses and forwards the request to DRAM for information retrieval, fails to teach " sending an instruction request to video memory, where a cache in the graphics pipeline sends the instruction request to the video memory” (see pg. 53, 54, 55, 56). Appellant further argues that memory data in Rivard, related to texel data, which fails to teach “sending an instruction request to video memory in a graphics pipeline, where a cache in the graphics pipeline

sends the instruction request to the video memory" (see pg. 54). Appellant further argues that the texture mode indicating a texture lookup, in addition to the memory request generator performing a memory request, as in Rivard, simply fails to suggest "... sending an instruction request to video memory, where a cache in the graphics pipeline sends the instruction request to the video memory" (see pg. 56). Appellant further argues that transferring graphical information into DRAM and outputting cached texture values, as in Rivard, fails to suggest "sending an instruction request to video memory, where a cache in the graphics pipeline sends the instruction request to the video memory" (see pg. 57).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a cache (texel cache system of the graphics pipeline) in a graphics pipeline sends the instruction request to the video memory (texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a

memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the texel cache system of the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texel cache system of the graphics pipeline for further processing).

- Regarding claim 30, appellant further argues "… Rivard's memory request to the DRAM fails to suggest an instruction request" (see pg. 53).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claim 30, appellant also argues that "… coordinating the arrival of memory data where the data is conveniently aligned in the texture map, as in Rivard, fails to teach sending an instruction request" (see pg. 54).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a cache (texel cache system of the graphics pipeline) in a graphics

pipeline sends the instruction request to the video memory (texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the texel cache system of the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texel cache system of the graphics pipeline for further processing).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions).

- Regarding claim 30, appellant further argues that generating and forwarding memory requests to DRAM, as in Rivard, fails to teach "sending an instruction request where instructions are received in response to the instruction request for storage in cache in the graphics pipeline" (see pg. 55, 58, 61). Appellant also argues that returning memory data to a cache data store, as in Rivard, fails to meet, "receiving instructions from the video memory in response to the instruction request for storage in the cache in

the graphics pipeline" (see pg. 58). Appellant further argues that forwarding memory requests to DRAM for information retrieval, as in Rivard, fails to teach "receiving instructions from the video memory in response to the instruction request for storage in the cache in the graphics pipeline" (see pg. 60).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In this case, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a cache (texel cache system of the graphics pipeline) in a graphics pipeline sends the instruction request to the video memory (texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a

memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the texel cache system of the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texel cache system of the graphics pipeline for further processing).

Rivard further teaches receiving instructions from the video memory in response to the instruction request for storage in the cache in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data to the texel cache system in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which stores this information and further sends it to the texture mapping stage).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated

instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit including a texture map data retrieval circuit that includes a cache control and cache memory unit (cache) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67, col. 7 lines 43-47; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit

executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that cache controller circuit 250 of the texture map retrieval circuit fetches the required texture map data from the local frame buffer for storage in cache memory 251). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 30, appellant further argues that performing memory requests before instruction information reaches the cache data store, as in Rivard, in addition to the disclosure of executing display instructions that may include texture data, as in Wang, fails to suggest “receiving instructions from the video memory in response to the instruction request for storage in the cache in the graphics pipeline” (see pg. 61). Appellant further argues “… it would not be obvious from the teachings of Rivard and Wang to receive instructions from the video memory in response to the instruction request for storage in the cache in the graphics pipeline" (see pg. 61).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a cache (texel cache system of the graphics pipeline) in a graphics pipeline sends the instruction request to the video memory (texel cache system comprising the memory request generator, pipeline latency elements, and cache data

store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the texel cache system of the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texel cache system of the graphics pipeline for further processing).

Rivard further teaches receiving instructions from the video memory in response to the instruction request for storage in the cache in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request

retrieves such data from DRAM; Rivard teaches to send back data to the texel cache system in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which stores this information and further sends it to the texture mapping stage).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions

are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit including a texture map data retrieval circuit that includes a cache control and cache memory unit (cache) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67, col. 7 lines 43-47; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that cache controller circuit 250 of the texture map retrieval circuit fetches the required texture map data from the local frame buffer for storage in cache memory 251). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 30, appellant argues "… the texture mapping stage is separate from the texture cache system, as disclosed in Rivard and thus fails to suggest sending an instruction request to video memory, where a cache in the graphics pipeline sends the instruction request to the video memory" (see pg. 52).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a cache (texel cache system of the graphics pipeline) in a graphics pipeline sends the instruction request to the video memory (texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the texel cache system of the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends

memory data based on the request to the texel cache system of the graphics pipeline for further processing).

- Regarding claim 18, appellant argues that the graphics subunit executing a series of display instructions found within a display list stored in computer memory as in Wang, does not teach "... a complete instruction set is received in response to the instruction request where the instructions are received from the video memory" (see pg. 62-64).

However, the examiner interprets that Rivard does not explicitly teach a complete instruction set is received in response to the instruction request. Therefore, the examiner brings in the Wang reference (col. 5 lines 43-67, col. 6 lines 1-47) that teaches a graphics subunit (texture module) of a graphics hardware system executing a series of display instructions (set of instructions) stored in computer memory (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the display list stored in the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data). The graphics subunit receives display instructions including texture data based on the supplied data and control signals (the examiner interprets that the claims do not specify how many instructions are needed to make a set complete; the examiner interprets that if the required operation can be performed from just the one received instruction, then it is considered as a complete set of instruction; the single display instruction received by the graphics from the sub-routine

process in response to the control signals correspond to a complete set of instruction; this instruction in the sub-routine is executed by the graphics subunit for rendering the concerned graphics primitive, and therefore it is considered as a complete set of instruction). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return a complete set of instructions as taught by Wang to the texture module of Rivard because this instruction set is needed to render the concerned graphics primitive (col. 5 lines 43-45 and lines 63-67).

- Regarding claims 19 and 20, appellant argues that the graphics subunit executing a series of display instructions found within a display list stored in computer memory as in Wang, does not teach "... a partial instruction set is received in response to the instruction request where the instructions are received from the video memory" (see pg. 65-66).

However, the examiner interprets that Rivard does not explicitly teach a partial instruction set is received in response to the instruction request. However, Wang (col. 5 lines 43-67, col. 6 lines 1-47) teaches a graphics subunit (texture module) of a graphics hardware system executing a series of display instructions (set of instructions) stored in computer memory. The graphics subunit receives display instructions including texture data based on the supplied data and control signals (the examiner interprets that the claims do not specify how many instructions are needed to make a set complete or partial; the examiner interprets that if the required operation can be performed from just the one received instruction, then it is considered as a complete set of instruction; however, the rendering process might need other instructions for rendering other

primitives, and in this case, the single received instruction is considered as a partial set of instruction; the single display instruction received by the graphics from the sub-routine process in response to the control signals correspond to a partial set of instruction; this instruction in the sub-routine is executed by the graphics subunit for rendering the concerned graphics primitive, but other instructions are needed for rendering other primitives, and therefore overall this single instruction is considered as a partial set of instruction). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 29, appellant argues that disclosing that a memory request is generated for all misses, as in Rivard, fails to teach “sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory” (see pg. 67, 71-72). Appellant further argues that performing a memory request before the address and instruction information reach the cache data store, as in Rivard, fail to teach "sending an instruction request” (see pg. 68). Appellant further argues that Rivard’s disclosure that the memory request generator generating a memory request and forwarding the request to DRAM for information retrieval, fails to teach "sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory” (see pg. 69, 70-71). Appellant further argues that storing memory data at the write

address, as in Rivard, fails to suggest "sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory" (see pg. 73). Appellant further argues that the texture mode indicating a texture lookup, in addition to the memory request generator performing a memory request, as in Rivard, simply fails to suggest "sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory" (see pg. 70). Appellant further argues that transferring graphical information into DRAM and outputting cached texture values, as in Rivard, fails to suggest "sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory" (see pg. 71).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In this case, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture

module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module.

However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture

mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module). Rivard also teaches a texture mapping stage as shown in Fig. 6 receiving data (instructions) from the rasterizer module of the pipeline and then receiving data (instructions) from the DRAM (it should be noted that the data has it's associated instructions; texture mapping stage in the graphics pipeline receives data from the rasterizer module to perform texturing/rendering operation; this data received from the rasterizer module is considered to be primary data). Therefore, the data received from the DRAM via the texel cache system is considered to be the additional data (Wang reference as shown below teaches that the instructions include data).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that

Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that the instructions and data returned by the display list to the graphics hardware system is considered additional instructions or data as the Rivard reference already showed that a rasterizer module passes on preliminary instructions or data to the texture module; it should also be noted that the graphics hardware system includes a 3D graphics subunit that includes a texture engine that is responsible for retrieving the texture map data for the polygon and mapping the texels of the texture data onto the pixels of the polygon; this texture engine of Wang is functional equivalent of the texture module as suggested

by Rivard). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

Although the combination of Rivard and Wang teach a method and system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, they do not explicitly teach controlling the texture module utilizing a shader module coupled thereto. However, AAPA teaches controlling the texture module utilizing a shader module coupled thereto (Fig.3; shader module is also coupled to the rasterizer module). Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to incorporate the shading module of AAPA into the texture module of Rivard and Wang because combination of shading module and texture module would enable a shading function to the graphic pipeline.

- Regarding claim 29, appellant further argues "… Rivard's memory request to the DRAM fails to suggest an instruction request" (see pg. 18).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claim 29, appellant argues "… the texture mapping stage is separate from the texture cache system, as disclosed in Rivard and thus fails to suggest "receiving additional instructions from the video memory in response to the instruction

request utilizing the texture module" (see pg. 74, 78). Appellant further argues that "… coordinating the arrival of memory data where the data is conveniently aligned in the texture map, as in Rivard, fails to teach receiving additional instructions" (see pg. 74). Appellant further argues that forwarding memory requests to DRAM for information retrieval, as in Rivard fails to teach "receiving additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 76). Appellant further argues that Rivard's disclosure that the memory request generator that generates memory requests that are forwarded to DRAM for information retrieval, fails to teach " receiving additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 77).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In this case, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency

elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present

invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module,

so the information/memory data is passed between the components of the texture module). Rivard also teaches a texture mapping stage as shown in Fig. 6 receiving data (instructions) from the rasterizer module of the pipeline and then receiving data (instructions) from the DRAM (it should be noted that the data has it's associated instructions; texture mapping stage in the graphics pipeline receives data from the rasterizer module to perform texturing/rendering operation; this data received from the rasterizer module is considered to be primary data). Therefore, the data received from the DRAM via the texel cache system is considered to be the additional data (Wang reference as shown below teaches that the instructions include data).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and

moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions). However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that the instructions and data returned by the display list to the graphics hardware system is considered additional instructions or data as the Rivard reference already showed that a rasterizer module passes on preliminary instructions or data to the texture module; it should also be noted that the graphics hardware system includes a 3D graphics subunit that includes a texture engine that is responsible for retrieving the texture map data for the polygon and mapping the texels of the texture data onto the pixels of the polygon; this texture engine of Wang is functional equivalent of the texture module as suggested by Rivard). Therefore, it would have been obvious to one of ordinary skill in art at the

time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 29, appellant further argues that performing memory requests before instruction information reaches the cache data store, as in Rivard, in addition to the disclosure of executing display instructions that may include texture data, as in Wang, fails to suggest "receiving additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 76). Appellant further argues "… it would not be obvious from the teachings of Rivard and Wang to receive additional instructions from the video memory in response to the instruction request utilizing the texture module" (see pg. 76-77).

However, the examiner interprets that Rivard (Fig. 6, Fig. 10, col. 4 lines 46-57, col. 6 lines 45-67, col. 7 lines 1-10) teaches sending an instruction request to video memory, where a texture module in a graphics pipeline sends the instruction request to the video memory (a texture mapping stage sends information to texel cache system; the texture mapping stage and the texel cache system together are considered as texture module, so the information is passed between the components of the texture module; texel cache system comprising the memory request generator, pipeline latency elements, and cache data store and memory data resolver forwards the memory requests/read requests to DRAM for information retrieval; the examiner interprets that Rivard discloses sending an instruction request to video memory by generating memory

request for all misses, and forwards the requests to DRAM for information retrieval). It is interpreted that an instruction request could be considered as merely a request because it is basically a request that requires DRAM to perform necessary operation. The examiner also interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM. It is further interpreted that when there is a miss, a memory requests/read requests is send to DRAM, which acts as an instruction to DRAM based on which DRAM performs a particular function of returning data for further processing. Therefore, it is reasonable to interpret that memory requests/read requests for all misses that are forwarded to DRAM are instruction requests based on which DRAM sends back information to the graphics pipeline for further processing (it should be noted that the examiner relies on the definition of instruction provided by dictionary.com, which defines instructions as a command given to a computer to carry out a particular operation and can contain data to be used in the operation; in this case, when the memory/read request is send to DRAM, DRAM sends memory data based on the request to the texture module of the graphics pipeline for further processing).

The examiner further interprets that Rivard does not explicitly teach texture mapping stage and texel cache system together functions as a texture module. However, it would have been obvious to one of ordinary skill in art at the time of present invention to combine texture mapping stage and texel cache system of Rivard to work together as a texture module (it should be noted that the examiner interprets texture mapping stage and texel cache system combined together will perform functions similar

to the claimed texture module, and therefore it would be obvious to modify the Rivard reference). The unity of diversity of parts would depend more upon the choice of the manufacturer, and the convenience and availability of the machines and tools necessary to construct the texture module, than on any inventive concept. One of ordinary skill in art, furthermore, would have expected applicant's invention to perform equally well with Rivard's reference that teaches to send and receive information/instruction from the texture mapping stage and texel cache system to the DRAM because using this components together will also result in sending and receiving instructions to and from DRAM. Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to modify Rivard to obtain the invention as specified in the claim.

Rivard further teaches receiving instructions from the video memory in response to the instruction request utilizing the texture module in the graphics pipeline (the examiner interprets that a DRAM can store instructions as data, and that the memory request retrieves such data from DRAM; Rivard teaches to send back data in some form in response to the memory request generated as a result of a miss; DRAM returns memory data to cache data store and memory data resolver component of texel cache system, which further sends this information to the texture mapping stage; the texture mapping stage and the texel cache system together are considered as texture module, so the information/memory data is passed between the components of the texture module). Rivard also teaches a texture mapping stage as shown in Fig. 6 receiving data (instructions) from the rasterizer module of the pipeline and then receiving data

(instructions) from the DRAM (it should be noted that the data has it's associated instructions; texture mapping stage in the graphics pipeline receives data from the rasterizer module to perform texturing/rendering operation; this data received from the rasterizer module is considered to be primary data). Therefore, the data received from the DRAM via the texel cache system is considered to be the additional data (Wang reference as shown below teaches that the instructions include data).

Rivard also teaches the arrival of memory data and associated instructions at the cache data store and memory data resolver component of the text cache system is coordinated by pipeline latency elements (col. 7 lines 4-7; memory data has associated instructions; it should be noted that depending on the DRAM design, there could be a constant latency to effect a page hit; however, in the instance when the instructions and data are combined together, no pipeline latency element will be required; therefore, when the Rivard reference is modified as taught by Wang below, the modified system might have the data and the instructions coordinated when arrived at cache data store and memory data resolver, and will not require the use of pipeline latency element; however, having the pipeline latency element will help to coordinate the instructions and data at their arrival at cache data store and memory data resolver, when there is a delay between the arrival of data and it's associated instructions; it should be noted that Rivard has this pipeline latency element in addition to the claimed limitation, and moreover, the examiner interprets that the modified system with pipeline latency element will still function as effectively and generate the desired results, even when there is no delay between the arrival of the data and it's associated instructions).

However, Rivard does not explicitly state that the data and it's associated instructions are returned by DRAM. Therefore, the examiner brings in the Wang reference that suggests a graphics subunit (texture module) in a graphics hardware system that supplies data and control signals to local frame buffer memory for executing a series of display instructions (Fig. 1, col. 5 lines 38-67; the computer memory includes the local frame buffer memory, which corresponds to the DRAM; based on the control signals send by the graphics hardware system, the frame buffer returns the polygon display instructions that includes the texture data, so that the graphics subunit executes the display instructions using this data; it should be noted that Wang teaches to return display instructions from the display list to the graphics hardware system for further processing, where the display instructions include texture data; it should be noted that the instructions and data returned by the display list to the graphics hardware system is considered additional instructions or data as the Rivard reference already showed that a rasterizer module passes on preliminary instructions or data to the texture module; it should also be noted that the graphics hardware system includes a 3D graphics subunit that includes a texture engine that is responsible for retrieving the texture map data for the polygon and mapping the texels of the texture data onto the pixels of the polygon; this texture engine of Wang is functional equivalent of the texture module as suggested by Rivard). Therefore, it would have been obvious to one of ordinary skill in art at the time of present invention to have the memory return instructions as taught by Wang to the texture module of Rivard because these instructions are needed to render the

graphics primitives to be displayed on the display device (col. 5 lines 43-45 and lines 63-67).

- Regarding claim 29, appellant argues that mere disclosure of texels resulting from a texture lookup, as in AAPA, fail to even suggest "sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory" (see pg. 68).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

- Regarding claim 29, appellant argues that AAPA, Rivard and Wang fail to suggest "sending an instruction request to video memory, where texture module couple to the shader module sends the instruction request to the video memory" (see pg. 72).

The above argument is similar in scope to the arguments discussed above. Please refer to the statements presented above in regards to the reasons provided for above arguments.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Jwalant Amin/

Examiner, Art Unit 2628

Conferees:

/Kee M Tung/

Supervisory Patent Examiner, Art Unit 2628

/XIAO M. WU/

Supervisory Patent Examiner, Art Unit 2628